Amendments to the Claims

The following Listing of Claims will replace all prior versions and listings of claims in the application.

Listing of Claims

Please amend claims 1-3, 5, 7, 11, and 77 and cancel claim 9 as follows:

1. (Currently amended) A method for handling plug-and-play events occurring at a client, said method comprising:

detecting ~~an~~ a first event notification of a plug-and-play event generated by a plug-and-play manager regarding a device in communication with a client communicating with a server over a network using a presentation-level protocol, the first event notification comprising a globally unique ID for the device, a vendor ID, a product ID, a device name, and an event type identifier;

intercepting said first event notification at an abstraction layer prior to receipt of the notification by an application or an operating system on the client;

redirecting said first event notification to the server from the client~~, before an operating system on the client can handle the plug-and-play event~~;

creating an association between (i) a virtual device in a user session on the server and (ii) the device in communication with the client;

generating a second event notification of a plug-and-play event in the user session for the virtual device, the second event notification including the created association;

receiving a command directed to the virtual device from an application executing in the user session on the server and in response to the second event notification;

determining the command is directed to the device in communication with the client, responsive to the created association;

forwarding the command to the client;

opening a virtual communication channel between the device in communication with the client and the application executing in the user session on the server; and

receiving at the client via the virtual communication channel, in response to the redirection of the first event notification, a command from the server, the command directed to

said device.

2. (Currently amended) The method of claim 1 wherein redirecting said first event notification further comprises:

generating a context identifier, said context identifier representing a virtual COM port;

binding the context identifier to the event notification; and

transmitting the bound context identifier and event notification to the server.

3. (Currently amended) The method of claim 1 wherein redirecting said first event notification includes redirecting said event notification via a virtual channel.

4. (Previously presented) The method of claim 1, wherein receiving a command from the server further comprises:

receiving from a server a command including a generated context identifier;

identifying the device using the context identifier; and

issuing a command to the identified device.

5. (Currently amended) The method of claim 1 wherein said first event notification is generated as a result of a device arrival.

6. (Original) The method of claim 5 wherein said command is an open command.

7. (Currently amended) The method of claim 1 wherein said first event notification is generated as a result of a device removal.

8. (Original) The method of claim 7 wherein said command is a close command.

9. (Cancelled)

10. (Original) The method of claim 1 wherein the device in communication with the client uses one of the USB (Universal Serial Bus) protocol, IEEE 1394 protocol, Bluetooth protocol, wi-fi protocol, wireless protocol, and infrared (IR) protocol to communicate with the client.

11. (Currently Amended) A method for handling plug-and-play events occurring at a client in communication with a server using a presentation-level protocol, said method comprising:

intercepting and redirecting from the client ~~an~~ a first event notification of a plug-and-play event generated by a plug-and-play (PnP) manager regarding a device in communication with the client to the server before an operating system or application on the client ~~can handle~~ receives the plug-and-play event, the first event notification comprising a globally unique ID for the device, a vendor ID, a product ID, a device name, and an event type identifier;

creating an association between a virtual device in a user session on the server and the device in communication with the client;

notifying an application program executing in the user session on ~~hosted by~~ the server of the occurrence of the first event notification by generating a second event notification in the user session of a plug-and-play event, the second event notification including the created association;

receiving, in response to notification of the occurrence of the first event notification, a command from the application program hosted by the server, the command directed to the virtual device; and

forwarding the command, responsive to the created association between the virtual device and the device in communication with the client.

12. (Currently amended) The method of claim 11 wherein the first event notification from the client is received over a virtual channel.

13. (Currently amended) The method of claim 11 wherein the first event notification includes a context identifier bound to the event notification, said context identifier representing a virtual COM port.

14. (Previously presented) The method of claim 11, further comprising: creating a server-unique name to identify the device connected to the client that generated the event notification, said server unique name used in mapping the client device to a specific session on the server established by the presentation level protocol.

15. (Previously presented) The method of claim 11 wherein notifying an application program further comprises: transmitting the event notification to applications communicating with the server within the session.

16. (Previously presented) The method of claim 11 wherein notifying an application program further comprises: transmitting the event notification only to applications communicating with the server which have previously registered a callback for a type of event causing the event notification.

17. (Currently amended) The method of claim 11 wherein said first event notification is generated as a result of a device arrival.

18. (Original) The method of claim 17 wherein said command is an open command.

19. (Currently amended) The method of claim 11 wherein said first event notification is generated as a result of a device removal.

20. (Original) The method of claim 19 wherein said command is a close command.

21-23. (Cancelled)

24. (Previously presented) The method of claim 1 further comprising:

emulating a plug-and-play event notification regarding a device in communication with the client.

25-27. (Cancelled)

28. (Previously presented) The method of claim 24 wherein the emulated event notification received from the client is received over a virtual channel.

29-67. (Cancelled)

68. (Currently amended) The method of claim 1 wherein detecting ~~an~~ a first event notification comprises:

detecting ~~an~~ a first event notification of a plug-and-play event regarding a device communicating with the client via a USB connection on the client.

69-76. (Cancelled)

77. (Currently Amended) A system for handling plug-and-play events occurring at a client, said ~~method~~ system comprising:

~~(a) providing~~ a client communicating with a server over a network using a presentation-level protocol;

~~(b)~~ means for detecting ~~an~~ a first event notification of a plug-and-play event generated by a plug-and-play (PnP) manager ~~notification~~ regarding a device in communication with the a client communicating with a server over a network using a presentation-level protocol, the first event notification comprising a globally unique ID for the device, a vendor ID, a product ID, a device name, and an event type identifier;

means for intercepting said first event notification at an abstraction layer prior to receipt of the notification by an application or an operating system on the client;

~~(c)~~ means for redirecting said first event notification to the server from the client~~, before an operating system on the client can handle the plug-and-play event~~;

means for creating an association between (i) a virtual device in a user session on the server and (ii) the device in communication with the client;

means for generating a second event notification of a plug-and-play event in the user session for the virtual device, the second event notification including the association;

means for receiving a command directed to the virtual device from an application executing in the user session on the server and in response to the second event notification;

means for determining the command is directed to the device in communication with the client, responsive to the created association;

means for forwarding the command to the client;

means for opening a virtual communication channel between the device in communication with the client and the application executing in the user session on the server; and

~~(d)~~ means for receiving at the client via the virtual communication channel, in response to the redirection of the first event notification, a command from the server, the command directed to said device.